

WHAT IS CLAIMED IS:

1 1. For use in a processor, a branch architecture for
2 limiting branch penalty without branch prediction
3 comprising:

4 a fetch-branch unit operating in parallel with a
5 decode unit and controlling retrieval of instructions for
6 the decode unit, wherein the fetch-branch unit, upon
7 detecting a branch instruction during one cycle,

8 initiates retrieval of at least one
9 sequential instruction from a location immediately
10 following a location of a last retrieved instruction
11 during one of a first cycle immediately following the
12 one cycle and a second cycle immediately following the
13 first cycle, and

14 initiates retrieval of at least one target
15 instruction from a target location for the branch
16 instruction during the other of the first cycle
17 immediately following the one cycle and the second
18 cycle immediately following the first cycle.

1 2. The branch architecture as set forth in Claim 1
2 wherein the fetch-branch unit resolves the branch
3 instruction and, upon resolving the branch instruction,
4 drops either the at least one sequential instruction or the
5 at least one target instruction.

1 3. The branch architecture as set forth in Claim 2
2 wherein the fetch-branch unit, upon resolving the branch
3 instruction, retrieves at least one instruction from a
4 location immediately following a location of a last
5 retrieved instruction within either the at least one
6 sequential instruction or the at least one target
7 instruction, depending upon whether a branch is taken.

1 4. The branch architecture as set forth in Claim 1
2 wherein the fetch-branch unit, upon detecting a branch
3 instruction during the one cycle, marks any fetched
4 instruction preceding the branch instruction with a regular
5 instruction type identifier, marks the branch instruction
6 with a branch instruction type identifier, and marks any
7 fetched instruction succeeding the branch instruction with
8 a sequential instruction type identifier.

1 5. The branch architecture as set forth in Claim 4
2 wherein the fetch-branch unit, upon not detecting a branch
3 instruction during the one cycle, marks all fetched
4 instruction with the regular instruction type identifier.

1 6. The branch architecture as set forth in Claim 1
2 wherein the fetch-branch unit marks the at least one
3 sequential instruction with a sequential instruction type
4 identifier.

1 7. The branch architecture as set forth in Claim 1
2 wherein the fetch-branch unit marks the at least one target
3 instruction with a target instruction type identifier.

1 8. A processor comprising:
2 at least one execution unit;
3 a decode unit; and
4 a branch architecture for limiting branch penalty
5 without branch prediction comprising:

6 a fetch-branch unit operating in parallel
7 with the decode unit and controlling retrieval of
8 instructions for the decode unit, wherein the fetch-
9 branch unit, upon detecting a branch instruction
10 during one cycle,

11 initiates retrieval of at least one
12 sequential instruction from a location
13 immediately following a location of a last
14 retrieved instruction during one of a first cycle
15 immediately following the one cycle and a second
16 cycle immediately following the first cycle, and

17 initiates retrieval of at least one
18 target instruction from a target location for the
19 branch instruction during the other of the first
20 cycle immediately following the one cycle and the
21 second cycle immediately following the first
22 cycle.

1 9. The processor as set forth in Claim 8 wherein the
2 fetch-branch unit resolves the branch instruction and, upon
3 resolving the branch instruction, drops either the at least
4 one sequential instruction or the at least one target
5 instruction.

1 10. The processor as set forth in Claim 9 wherein the
2 fetch-branch unit, upon resolving the branch instruction,
3 retrieves at least one instruction from a location
4 immediately following a location of a last retrieved
5 instruction within either the at least one sequential
6 instruction or the at least one target instruction,
7 depending upon whether a branch is taken.

1 11. The processor as set forth in Claim 9 wherein the
2 fetch-branch unit, upon detecting a branch instruction
3 during the one cycle, marks any fetched instruction
4 preceding the branch instruction with a regular instruction
5 type identifier, marks the branch instruction with a branch
6 instruction type identifier, and marks any fetched
7 instruction succeeding the branch instruction with a
8 sequential instruction type identifier.

1 12. The processor as set forth in Claim 11 wherein
2 the fetch-branch unit, upon not detecting a branch
3 instruction during the one cycle, marks all fetched
4 instruction with the regular instruction type identifier.

1 13. The processor as set forth in Claim 8 wherein the
2 fetch-branch unit marks the at least one sequential
3 instruction with a sequential instruction type identifier.

1 14. The processor as set forth in Claim 8 wherein the
2 fetch-branch unit marks the at least one target instruction
3 with a target instruction type identifier.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

1 15. For use in a processor, a method of processing
2 branch instructions without branch prediction comprising:

3 operating a fetch-branch unit in parallel with a
4 decode unit to control retrieval of instructions for the
5 decode unit; and

6 upon detecting a branch instruction during one
7 cycle,

8 initiating retrieval of at least one
9 sequential instruction from a location immediately
10 following a location of a last retrieved instruction
11 during one of a first cycle immediately following the
12 one cycle and a second cycle immediately following the
13 first cycle, and

14 initiating retrieval of at least one target
15 instruction from a target location for the branch
16 instruction during the other of the first cycle
17 immediately following the one cycle and the second
18 cycle immediately following the first cycle.

1 16. The method as set forth in Claim 15 further
2 comprising:

3 resolving the branch instruction; and
4 upon resolving the branch instruction, dropping
5 either the at least one sequential instruction or the at
6 least one target instruction.

1 17. The method as set forth in Claim 16 further
2 comprising:

3 upon resolving the branch instruction, retrieving
4 at least one instruction from a location immediately
5 following a location of a last retrieved instruction within
6 either the at least one sequential instruction or the at
7 least one target instruction, depending upon whether a
8 branch is taken.

1 18. The method as set forth in Claim 15 further
2 comprising:

3 upon detecting a branch instruction during the
4 one cycle,

5 marking any fetched instruction preceding
6 the branch instruction with a regular instruction type
7 identifier,

8 marking the branch instruction with a branch
9 instruction type identifier, and

10 marking any fetched instruction succeeding
11 the branch instruction with a sequential instruction
12 type identifier.

1 19. The method as set forth in Claim 18 further
2 comprising:

3 upon not detecting a branch instruction during
4 the one cycle, marking all fetched instruction with the
5 regular instruction type identifier.

```

5           marking the at least one target instruction with
6           a target instruction type identifier.

```

Station	Time	Lat	Long	Alt	Wind	Temp	Humid	Cloud	Vis	Bar	Notes
10101	0100	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10102	0200	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10103	0300	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10104	0400	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10105	0500	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10106	0600	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10107	0700	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10108	0800	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10109	0900	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10110	1000	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10111	1100	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10112	1200	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10113	1300	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10114	1400	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10115	1500	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10116	1600	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10117	1700	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10118	1800	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10119	1900	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10120	2000	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10121	2100	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10122	2200	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10123	2300	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10124	2400	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10125	2500	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10126	2600	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10127	2700	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10128	2800	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10129	2900	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear
10130	3000	33° 15'	110° 10'	100	0	55	85	0	10	30.0	Clear